

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет
імені Івана Пулюя

Кафедра автоматизації технологічних
процесів та виробництв

Методичні вказівки
для виконання лабораторної роботи № 8
“Вивід інформації через послідовний порт
мікроконвертера ADuC842”
з курсу “Розробка систем керування на основі
OMEOM”

Тернопіль 2016

Методичні вказівки для виконання лабораторної роботи № 8 «Вивід інформації через послідовний порт мікроконвертера ADuC842» з курсу «Розробка систем керування на основі OMEOM».

Методичні вказівки розглянуті і схвалені кафедрою «Автоматизація технологічних процесів і виробництв», протокол № 4 від 21.11.2016 р.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р.,
асистент Пісьціо В.П.

Лабораторна робота № 8

Вивід інформації через послідовний порт мікроконвертора ADuC842

1. Мета роботи

1.1. Вивчити особливості роботи послідовних портів мікроконвертора.

2. Попередня підготовка до роботи

2.1. Вивчити схему послідовного порту мікроконтролерів.

2.2. Вивчити схеми узгодження логічних рівнів послідовних портів мікроконтролера і персонального комп'ютера.

2.3. Вивчити особливості використання послідовного порту мікроконтролера.

2.4. Скласти програму, що виводить напис на екран комп'ютера.

3. Методичні вказівки по роботі

Для зв'язку мікроконтролера і ПК найчастіше використовується послідовний порт.

Це пов'язано з двома причинами.

Перша - це мінімальна кількість проводів, що вимагається для зв'язку між цими пристроями.

Друга - це висока захищеність послідовного порту від пошкоджень при підключенні різних пристроїв.

Однак, при роботі з послідовним портом мікроконтролера виникає ряд труднощів. Основна складність полягає в тому, що послідовний порт мікроконтролера використовує логічні TTL рівні (рівнем логічного нуля вважають напругу нижче 0.4 В, рівнем логічної одиниці - напругу більше 2,4 В).

Послідовний порт комп'ютера при передачі логічного нуля формує напругу +10 В, а при передачі логічної одиниці - напругу -10 В.

Для узгодження логічних рівнів COM-порту комп'ютера і послідовного порту мікроконтролера в даний час використовуються спеціалізовані мікросхеми.

3.1. Налаштування послідовного порту мікроконтролера

Через універсальний послідовний порт мікроконтролера здійснюються прийом і передача інформації, представлена в послідовному коді (молодшими бітами вперед).

Наявність буферного регістра приймача дозволяє поєднувати операцію читання раніше прийнятого байту з прийомом чергового. Але якщо до моменту закінчення прийому байту попередній не був зчитаний з регістра SBUF, то його буде втрачено.

Роботою послідовного порту управляють три регістри:

- Регістр управління / статусу приймача **SCON**.
- Біт SMOD регістра управління потужністю **PCON**.
- Буферний регістр приймача **SBUF**.

Послідовний порт може працювати в **чотирьох різних режимах**, але для зв'язку мікроконтролера з комп'ютером підходить тільки режим 1.

У цьому режимі можуть бути передані через зовнішній вивід **TxD** або прийняті через зовнішній вивід **RxD** 8 бітів даних. При прийомі, стоп-біт записується в флажок RB8 регістра **SCON**.

Швидкість передачі послідовного порту в режимі 1 задається в AduC842 таймером T3 або таймером T1 чи T2.

Послідовний порт починає передачу після будь-якої команди, яка здійснює запис інформації в регістр **SBUF**.

Прийом починається з приходом старт-біту, якщо в біт **REN** регістра **SCON** записана логічна одиниця.

Управління режимом роботи приймача здійснюється через регістр управління послідовного порту **SCON**. Цей реєстр містить не тільки керуючі біти, що визначають режим

роботи послідовного порту, але і дев'ятий біт прийнятих або переданих даних (RB8 і TB8) і біти переривання приймача (RI і TI).

Біт TI встановлюється апаратно після завершення передачі одного байту. При передачі декількох байтів цей біт повинен порівнюватися з «1», і тільки після порівняння можна передавати наступний байт.

Перед передачею чергового байту біт TI слід програмно обнулити. Формат і адреси бітів регістра SCON наведено на наступному рисунку 1:

Адреса	Значення старших розрядів				Значення молодших розрядів				
98H	SM0 SCON.7	SM1 SCON.6	SM2 SCON.5	REN SCON.4	TB8 SCON.3	RB8 SCON.2	TI SCON.1	RI SCON.0	SCON
	9Fh	9Eh	9Dh	9Ch	9Bh	9Ah	99h	98h	

Рисунок 1. Формат регістра SCON

Функціональне призначення бітів регістра управління/статусу приймача SCON приведені в наступній таблиці:

Символ	Позиція	Ім'я та призначення		
SM0	SCON.7	Біти управління режимом роботи прийомопередавача. Встановлюються/скидаються програмно		
SM1	SCON.6			
		SM0	SM1	Режим роботи прийомопередавача
		0	0	Зсувний регістр розширення вводу/виводу
		0	1	8 бітовий прийомопередавач, змінна швидкість передачі
		1	0	9 бітовий прийомопередавач. Фіксована швидкість передачі
		1	1	9 бітовий прийомопередавач, змінна швидкість передачі
SM2	SCON.5	Біт управління режимом прийомопередавача. Встановлюється програмно для заборони прийому повідомлення, в якому дев'ятий біт має значення 0		
REN	SCON.4	Біт дозволу прийому. Встановлюється/скидається програмно для дозволу/заборони прийому послідовних даних		
TB8	SCON.3	Передача біту 8. Встановлюється/скидається програмно для задання дев'ятого передаваного біту в режимі 9-бітового передавача		
RB8	SCON.2	Прийом біту 8. Встановлюється/скидається апаратно для фіксації дев'ятого прийманого біту в режимі 9-бітового приймача		
TI	SCON.1	Флаг переривання передавача. Встановлюється апаратно при закінченні передачі байту. Скидається програмно після обслуговування переривання		
RI	SCON.0	Флаг переривання приймача. Встановлюється апаратно при прийомі байту. Скидається програмно після обслуговування переривання		

Для того, щоб включити **перший режим роботи** послідовного порту, досить виконати наступні команди:

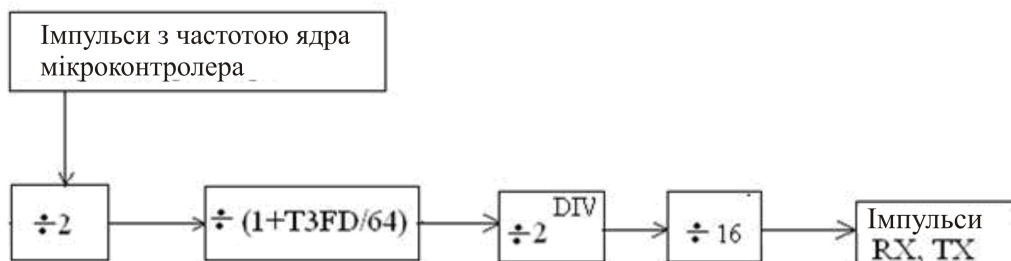
CLR SCON, 7	; вибрати 8-ми бітний режим роботи послідовного порту,
SETB SCON, 6	; для чого встановити «1» в шостий біт, і «0» - в сьомий
SETB SCON, 4	; дозволити роботу приймача (REN=1)
CLR SCON, 0	; дозволити запис першого байту в передавач, для чого
	; очистити біт RI
SETB SCON, 1	; встановити «1» в перший біт

Це програмування можна здійснити однією командою:

MOV SCON, # 52h	; 8 bit UART, дозвіл роботи приймача і передавача
-----------------	---

Для задання точних часових інтервалів при роботі послідовного інтерфейсу UART в ADuC842 є спеціальний таймер **T3**.

Часові інтервали, що формуються таймером, визначаються двома регістрами спеціальних функцій: **T3FD** і **T3CON**. Формування імпульсів, що синхронізують роботу UART, визначається наступною схемою:



Зі схеми видно, що таймер T3 має кілька подільників частоти (подільники на 2, на $(1 + T3FD / 64)$, на 2 в ступені DIV, на 16).

В результаті отримується частота імпульсів на лініях передавача TX і приймача RX. Величина DIV визначається трьома молодшими розрядами регістра T3CON.

Біт	Мнемоніка	Призначення
T3CON.7	T3BAUDEN	Біт включення Таймера 3. Якщо T3BAUDEN=1, то Таймер 3 генерує опорну частоту UART, при цьому біти PCON.7,T2CON.4 і T2CON.5 ігноруються. Якщо T3BAUDEN=0, то опорна частота UART генерується відповідно до стандартної реалізації в ядрі 8052.
T3CON.6	—	Біти, зарезервовані для наступного використання.
T3CON.5	—	
T3CON.4	—	
T3CON.3	—	
T3CON.2		Цілочислений подільник <i>DIV</i>
T3CON.1		DIV2 DIV1 DIV0 DIV
T3CON.0		0 0 0 0
		0 0 1 1
		0 1 0 2
		0 1 1 3
		1 0 0 4
		1 0 1 5
	1 1 0 6	
	1 1 1 7	

Таким чином, швидкість передачі послідовного порту можна визначити по наступній формулі:

$$ActualBaudRate = \frac{2 \cdot f_{osc}}{2^{DIV-1} \cdot (T3FD + 64)}$$

де ActualBaudRate – швидкість передачі даних
 f_{osc} – частота ядра.

При включенні живлення мікроконтролера частота $f_{osc} = 2.097152$ МГц (визначається трьома молодшими розрядами регістра **PLLCON**, значення яких для вказаної частоти становлять 011b=03h).

Якщо задати $T3FD = 2Dh = 45(10)$, а $DIV = 3$, то швидкість передачі буде дорівнювати:

$$\text{ActualBaundRate} (2 \cdot 2.097152) / (4 \cdot (45 + 64)) = 0.0096199 \text{ МГц},$$

тобто, приблизно 9600 бітів за секунду.

3.2. Приклад програми роботи з послідовним портом

; Програма, яка приймає байт від комп'ютера, виводить його на світлодіодну лінійку і відсилає назад в COM-порт комп'ютера

```
org 0000h                ;
call uart_ini            ; ініціалізація aduc842
                        ; приймання байту з виводом на світлодіодну лінійку
loop: jnb scon.0,$        ; очікування байту
mov a,sbuf               ; прийом байту
clr scon.0               ; скидання флага приймача
call svdisp              ; вивід на світлодіоди
                        ; передача байту назад до комп'ютера через COM-порт
mov sbuf, a              ; передача байту
jnb scon.1, $            ; очікування кінця передачі
clr scon.1               ; скидання флага передавача
sjmp loop                ; кінець програми

                        ; програма UART_INI для ADuC842 (швидкість 9600 бод)
pllcon data 0D7h         ; оголошення змінної PLLCON як регістра з адресою 0D7
t3con data 09Eh          ; оголошення змінної T3CON як регістра з адресою 09E
t3fd data 09Dh           ; оголошення змінної T3FD як регістра з адресою 09D

uart_ini:                ; програмування послідовного інтерфейсу
mov pllcon, #03h         ; визначення регістром PLLCON частоти ядра f=2.097152 MHz
mov t3con, #83h          ; T3CON=83h (DIV=3)
mov t3fd, #2dh           ; T3FD =2dh,
mov scon, #52h           ; 8 bit UART, дозвіл роботи приймача і передавача
ret
end
```

4. Завдання до роботи в лабораторії

4.1. Написати програму, що виводить Ваше прізвище, записану латинськими літерами, через послідовний порт.

4.2. Вибрати мікроконтролер AduC842, встановити галочку навпроти Create HEX file (інакше не буде створено hex-файл, навіть якщо немає помилок).

4.3. Скопіювати екран з інтерфейсом Keil з результатами виконання програми в звіт.

4.4. Виконайте покрокове налагодження програми з використанням кнопки F11.

На кожному кроці виконання програми запишіть значення використовуваних регістрів відповідно до таблиці 2.

Таблиця 2. Результати виконання команд

№	Команда	Код	Виконувана операція	Вміст використовуваних регістрів і комірок пам'яті до і після виконання		Пояснення
				До	Після	
1				A/00 PC/00 PSW/00	A/F2 PC/01 PSW/01	
2
...
...

5. Зміст звіту

5.1. Мета роботи.

5.2. Текст програми.

5.3. Текст hex-файлу

5.4. Копія екрану з інтерфейсом Keil з результатами виконання програми.

5.5. Висновки про виконану лабораторну роботу.